UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/965,518	09/25/2001	Nagabhushana T. Sindhushayana	000400	3313

23696 7590 10/17/2006

QUALCOMM INCORPORATED
5775 MOREHOUSE DR.
SAN DIEGO, CA 92121

EXAMINER	
TORRES, JUAN A	
ART UNIT	PAPER NUMBER
2611	

DATE MAILED: 10/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

ℙ

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/965,518
Filing Date: September 25, 2001
Appellant(s): SINDHUSHAYANA ET AL.

Larry Moskowitz
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 08/04/2006 appealing from the Office action mailed 06/23/2006

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

The statement of the status of claims contained in the brief is correct.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

A substantially correct copy of appealed claim 6 appears on page 21 of the Appendix to the appellant's brief. The minor errors are as follows: The claim 6 presented in the Appeal brief correspond to the claim 6 proposed amendment after final that was not entered (see underling <u>at</u> in the last line of claim 6).

(8) Evidence Relied Upon

Applicants Admitted Prior Art

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

Claims 1-3, 6, 7, 9-14, 17, 18 and 20-24 are rejected under 35 U.S.C. 102(a) as being anticipated by admitted prior art in FIG. 5).

As per claim 1, admitted prior art in FIG. 5 shows a method for decoding a sequence of turbo encoded data symbols transmitted over a channel comprising: updating channel nodes R_x, R_y and R_z based on a received channel output (in FIG. 5 R_x is block 501 input 541 and 542; R_y is block 501 input 542 and R_z is block 502 input 540); initializing outgoing messages from symbol nodes X_i, Y_i and Z_k where the symbol nodes X_i, Y_i and Z_k are in communication with the channel nodes R_x, R_y and R_z (X_i, is block 501 output 550; Y_i is output of block 520 line 542 and Z_k is output of block 520 line 540); and triggering updates of computational nodes C and D, (computational node C is block 501 and computational node D is block 502) associated with different instances of time, in accordance with a triggering schedule, where a computational node C_i is in communication with the symbol nodes X_i, and Y_i and a computational node D_k is in communication with the symbol nodes X_i and Z_k.(FIG. 5 input of block 501 – C – have inputs X_i, and Y_i and block 502 – D – have inputs X_i and Z_k); where the triggering schedule includes triggering all the computational nodes C and D at different instances of time essentially concurrently for each decoding iteration (block 501 (computational node C), is the first decoder with inputs, essentially, X (information data) and Y (fist parity), when the first decoder finish, pass the output to block 502 (computational node D), that at a different instant in time, essentially concurrently, with inputs X (information data) and Z (second parity, or parity from de second encoder) produces the first estimation of the data in iteration zero).

As per claim 2, admitted prior art in FIG. 5 teaches claim 1, admitted prior art in FIG. 5 also teaches decoding a sequence of turbo code where the computational node C_i is in communication with state nodes S_i and S_{i-1}, associated with a first constituent code (FIG. 5 input of block 501 – C – have inputs X_i, with be related to S_i and output X_i will be related with S_{i-1}), and the computational node D_k is in communication with state node σ_k and σ_{k-1} associated with a second constituent code, where the first and second constituent codes are associated with a turbo code in the communication system used for encoding the sequence of encoded data symbols (FIG. 5 block 501 – C – have inputs X_i, with be related to S_i and output X_i will be related with S_{i-1} and block 502 – D – have inputs X_k, with be related to σ_{k-1} and output X_k will be related with σ_k) .

As per claim 3, admitted prior art in FIG. 5 teaches claim 1, admitted prior art in FIG. 5 also teaches accepting a value of symbol X_i at the symbol node X_i as a decoded value of symbol X_i after at least one iteration of the triggering updates of the computational nodes C and D (FIG. 5 output of block 501 line 550 after the first cycle).

As per claim 6, admitted prior art in FIG. 5 shows a method for decoding a sequence of turbo encoded data symbols transmitted over a channel comprising: updating channel nodes R_x, R_y and R_z based on a received channel output (in FIG. 5 R_x is block 501 input 541 and 542; R_y is block 501 input 542 and R_z is block 502 input 540); initializing outgoing messages from symbol nodes X_i, Y_i and Z_k where the symbol nodes X_i, Y_i and Z_k are in communication with the channel nodes R_x, R_y and R_z (X_i, is block 501 output 550; Y_i is output of block 520 line 542 and Z_k is output of block 520 line 540); and triggering updates of computational nodes C and D, (computational node C is block 501 and computational node D is block 502) associated with different instances of time, in accordance with a triggering schedule, where a computational node C_i is in communication with the symbol nodes X_i, and Y_i and a computational node D_k is in communication with the symbol nodes X_i and Z_k.(FIG. 5 input of block 501 – C – have inputs X_i, and Y_i and block 502 – D – have inputs X_i and Z_k); partitioning the computational node C at time instances C_0, C_1, C_2, … , C_N into at least two subsets, where the triggering schedule includes triggering updates of computational nodes C in a sequence at different time instances in each subset (FIG. 5 input 541 of block 501 wait until block 502 produces its output 560 and that output is deinterleaved by block 531 and this process is repeated until a determined number of iterations that define a number of subsets) and where the triggering of computational node C at different instances in the at least two subsets occurs concurrently (block 501 (computational node C), is the first decoder with inputs, essentially, X (information data) and Y (fist parity), when the first decoder finish, pass the output to block 502 (computational node D), that at a different instant in time, essentially concurrently, with inputs X (information data) and Z (second parity, or parity from de second encoder) produces the first estimation of the data in iteration zero. To produce the first iteration (here the term turbo) the information from the decoder 502 have to go to the first decoder 501 as extrinsic information, and the first decoder have to run again producing a new output, for this reason, the computational node C have to have at least two subsets, to produce the first iteration. Without a first iteration the turbo process doesn't take place).

As per claim 7, admitted prior art in FIG. 5 teaches claim 6, admitted prior art in FIG. 5 also teaches determining the sequence at different time instances in each subset for the triggering updates FIG. 5 input 541 of block 501 wait until block 502 produces its output 560 and that output is deinterleaved by block 531 and this process is repeated until a determined number of iterations that define a number of subsets that happen a different time instances, each iteration wait for the previous iteration).

As per claim 9, admitted prior art in FIG. 5 teaches claim 1, admitted prior art in FIG. 5 also teaches that the at least two subsets of computational node C at different time instances C_0, C_1, C_2, ... , C_N have at least one common computational node time instance FIG. 5 input 541 of block 501 wait until block 502 produces its output 560 and that output is deinterleaved by block 531 and this process is repeated until a determined number of iterations that define a number of subsets that happen a different time instances, each iteration wait for the previous iteration. The block 501 will hold the value Y_i of waiting for the next value of $X_{i,}$ X_{i+1} that will be in a common computational node time instance).

As per claim 10, admitted prior art in FIG. 5 teaches claim 6, admitted prior art in FIG. 5 also teaches partitioning computational node D at different time instances D_0, D_1, D_2, ... , D_N into at least two subsets, where the triggering schedule includes triggering computational nodes D at different time instances in a sequence in each subset (FIG. 5 input 532 of block 502 wait until block 501 produces its output 550 and that output is interleaved by block 530 and this process is repeated until a determined number of iterations that define a number of subsets).

As per claim 11, admitted prior art in FIG. 5 teaches claim 10, admitted prior art in FIG. 5 also teaches determining the sequence at different time instances in each subset for the triggering updates (FIG. 5 input 532 of block 502 wait until block 501 produces its output 550 and that output is interleaved by block 530 and this process is repeated until a determined number of iterations that define a number of subsets that happen a different time instances, each iteration wait for the previous iteration).

As per claim 12, admitted prior art in FIG. 5 teaches claim 10, admitted prior art in FIG. 5 also teaches that the triggering of computational node D at different time instance in the least two subsets occurs concurrently (FIG. 5 input 532 of block 502 wait until block 501 produces its output 550 and that output is interleaved by block 530 and this process is repeated until a determined number of iterations that define a number of subsets that happen a different time instances, each iteration wait for the previous iteration. The block 502 will hold the value Z_k of waiting for the next value of X_k, X_{k+1}).

As per claim 13, admitted prior art in FIG. 5 teaches claim 10, admitted prior art in FIG. 5 also teaches that the subsets of computational node D at time instances D_0, D_1, D_2, ... , D_N have at least one common computational node time instance (FIG. 5 input 532 of block 502 wait until block 501 produces its output 550 and that output is interleaved by block 530 and this process is repeated until a determined number of iterations that define a number of subsets that happen a different time instances, each iteration wait for the previous iteration. The block 502 will hold the value Z_k of waiting for the next value of X_k, X_{k+1} that will be in a common computational node time instance).

As per claim 14, admitted prior art in FIG. 5 teaches claim 1, admitted prior art in FIG. 5 also teaches that the updating includes summing incoming messages to produce an output message, and outputting the output message for updating (FIG. 5 input 541 of block 501 X_i 2nd estimation will produce an updated output in 550 that will be the third estimation).

As per claim 17, admitted prior art in FIG. 5 teaches claim 1, admitted prior art in FIG. 5 also teaches that the sequence of data includes "N" number of symbols, each symbol in the sequence is identified by either a subscript "i" or "k," and the subscript "i" and "k" are references to time instances in the decoding process (FIG. 5 subscript "i" is input to block 501 related with not-interleaved data and subscript "k" is input to block 502 related with interleaved data).

As per claim 18, admitted prior art in FIG. 5 shows a method for decoding a sequence of turbo encoded data symbols transmitted over a channel comprising: updating channel nodes R_x, R_y and R_z based on a received channel output (in FIG. 5 R_x

is block 501 input 541 and 542; R_y is block 501 input 542 and R_z is block 502 input 540); initializing outgoing messages from symbol nodes X_i, Y_i and Z_k where the symbol nodes X_i, Y_i and Z_k are in communication with the channel nodes R_x, R_y and R_z (X_i, is block 501 output 550; Y_i is output of block 520 line 542 and Z_k is output of block 520 line 540); state nodes S_i and S_{i-1} associated with a first constituent code in a turbo code (in FIG. 5 block 501 output 550 and inputs 541 and 542); state nodes σ_k and σ_{k-1} associated with a second constituent code in the turbo code (in FIG. 5 block 502 output 560 and inputs 540 and 532); a computational node C_i in communication with the symbol nodes X_i and Y_i (computational node C_i is block 501); and a computational node D_k in communication with the symbol nodes X_i and Z_k (computational node D_k is block 502), where the computational node C_i is in communication with the state nodes S_i and S_{i-1} (in FIG. 5 block 501 output 550 and inputs 541 and 542) and the computational node D_k is in communication with the state nodes σ_k and σ_{k-1} (in FIG. 5 block 502 output 560 and inputs 540 and 532); a computational node C_{i+1} in communication with the state node S_i (in FIG. 5 block 501 inputs 541 and 542); a computational node C_{i-1}, in communication with the state node S_{i-1} (in FIG. 5 block 501 output 550); a computational node D_{k+1} in communication with the state node σ_k (in FIG. 5 block 502 inputs 540 and 532); and a computational node D_{k-1} in communication with the state node σ_{k+1} (in FIG. 5 block 502 output 560), where computational nodes C and D at different time instances are configured for updates in accordance with a update triggering schedule and the update triggering schedule including concurrent triggering of each node of a first plurality of the computational nodes C, and concurrent trickery of each node of a second plurality of computational nodes D (block 501 (computational node C), is the first decoder with inputs, essentially, X (information data) and Y (fist parity), when the first decoder finish, pass the output to block 502 (computational node D), that at a different instant in time, essentially concurrently, with inputs X (information data) and Z (second parity, or parity from de second encoder) produces the first estimation of the data in iteration zero. To produce the first iteration (here the term turbo) the information from the decoder 502 have to go to the first decoder 501 as extrinsic information, and the first decoder have to run again producing a new output, for this reason, the computational node C have to have at least two subsets, to produce the first iteration. Without a first iteration the turbo process doesn't take place).

As per claim 20, admitted prior art in FIG. 5 teaches claim 18, admitted prior art in FIG. 5 also teaches that the update triggering schedule includes triggering updates in a sequence in a partitioned computational nodes C_0, C_1, C_2, ... , C_N of at least two subsets and in a sequence in a partitioned computational nodes D_0, D_1, D_2, ... , D_N of at least two subsets (FIG. 5 input 541 of block 501 wait until block 502 produces its output 560 and that output is deinterleaved by block 531 and this process is repeated until a determined number of iterations that define a number of subsets, input 532 of block 502 wait until block 501 produces its output 550 and that output is interleaved by block 530 and this process is repeated until a determined number of iterations that define a number of subsets).

As per claim 21, admitted prior art in FIG. 5 teaches claim 18, admitted prior art in FIG. 5 also teaches that the sequence of data includes "N" number of symbols, where each symbol in the sequence is identified by either a subscript "i" or "k" corresponding to the subscripts used for the state nodes and the computational nodes (FIG. 5 subscript “i” is input to block 501 related with not-interleaved data and subscript “k” is input to block 502 related with interleaved data).

As per claim 22, admitted prior art in FIG. 5 shows a processor configured for decoding a sequence of turbo encoded data symbols for communication over a channel comprising: channel nodes R_x, R_y and R_z for receiving channel output (in FIG. 5 R_x is block 501 input 541 and 542; R_y is block 501 input 542 and R_z is block 502 input 540); symbol nodes X_i, Y_i and Z_k in communication with the channel nodes R_x, R_y and R_z ($X_{i,}$ is block 501 output 550; Y_i is output of block 520 line 542 and Z_k is output of block 520 line 540); state nodes S_i and S_{i-1} associated with a first constituent code in a turbo code (in FIG. 5 block 501 output 550 and inputs 541 and 542); state nodes σ_k and σ_{k-1} associated with a second constituent code in the turbo code (in FIG. 5 block 502 output 560 and inputs 540 and 532); a computational node C_i in communication with the symbol nodes X_i and Y_i (computational node C_i is block 501); and a computational node D_k in communication with the symbol nodes X_k and Y_k (computational node D_k is block 502), where the computational node C_i is in communication with the state nodes S_i and S_{i-1} (in FIG. 5 block 501 output 550 and inputs 541 and 542) and the computational node D_k is in communication with the state nodes σ_k and σ_{k-1} (in FIG. 5 block 502 output 560 and inputs 540 and 532); a computational node C_{i+1} in communication with the state node S_i (in FIG. 5 block 501 inputs 541 and 542); a computational node C_{i-1} in communication with the state node S_{i-1} (in FIG. 5 block 501 inputs 541 and 542); a computational node D_{k+1}in communication with the state node σ_k (in FIG. 5 block 502 inputs 540 and 532); and a computational node D_{k-1} in communication with the state node σ_{k+1} (in FIG. 5 block 502 output 560), where computational nodes C and D at different time instances are configured for updates in accordance with a update triggering schedule and the update triggering schedule including concurrent triggering of each node of a first plurality of the computational nodes C, and concurrent trickery of each node of a second plurality of computational nodes D (block 501 (computational node C), is the first decoder with inputs, essentially, X (information data) and Y (fist parity), when the first decoder finish, pass the output to block 502 (computational node D), that at a different instant in time, essentially concurrently, with inputs X (information data) and Z (second parity, or parity from de second encoder) produces the first estimation of the data in iteration zero. To produce the first iteration (here the term turbo) the information from the decoder 502 have to go to the first decoder 501 as extrinsic information, and the first decoder have to run again producing a new output, for this reason, the computational node C have to have at least two subsets, to produce the first iteration. Without a first iteration the turbo process doesn't take place).

As per claim 23, admitted prior art in FIG. 5 teaches claim 22, admitted prior art in FIG. 5 also teaches that the update triggering schedule includes triggering updates of the computational nodes C and D in a sequence of $C_0, C_1, C_2, \ldots, C_N, C_{N-1}, C_{N-2}, C_{N-3}, \ldots C_2, C_1, C_0, D_0, D_1, D_2, \ldots, D_N, D_{N-1}, D_{N-2}, D_{N-3}, \ldots D_2, D_1, D_0$ (FIG. 5 input 532 of block 502 wait until block 501 produces its output 550 and that output is interleaved by block 530 and this process is repeated until a determined number of iterations).

As per claim 24, admitted prior art in FIG. 5 teaches claim 22, admitted prior art in FIG. 5 also teaches that the sequence of data includes "N" number of symbols, where each symbol in the sequence is identified by either a subscript "i" or "k" corresponding to the subscripts used for the state nodes and the computational nodes (FIG. 5 subscript "i" is input to block 501 related with not-interleaved data and subscript "k" is input to block 502 related with interleaved data).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 15 and 16 rejected under 35 U.S.C. 103(a) as being unpatentable over admitted prior art as applied to claim 1 above, and further in view of Xu (US 20010052104).

As per claim 15, admitted prior art discloses claim 1. Admitted prior art in FIG. 5 discloses a method for decoding a sequence of turbo code where the updating of the channel nodes R_x, R_y and R_z based on the received channel output includes receiving at the channel node R_x the channel output associated with a symbol X_i; receiving at the channel node R_y the channel output associated with a symbol Y_i; receiving at the channel node R_z the channel output associated with a symbol Y_k; passing from the channel node R_x a next value of the symbol X_i, based on the received channel output, to the symbol node X_i; passing from the channel node R_y a next value of the symbol Y_i, based on the received channel output, to the symbol node Y_i; and passing from the channel node R_z a next value of the symbol Z_k, based on the received channel output, to the symbol node Z_k. FIG. 5 doesn't teach that the next value is a representation of the likelihood of the value, but this is inhering in the process of turbo decoding, a new update in a value will represent the likelihood of this value in comparison with the previous value, this is very well known in turbo decoding process and Xu teaches the process of passing from the channel node R_x a likelihood of the symbol X_i, based on the received channel output, to the symbol node X_i; passing from the channel node R_y a likelihood of the symbol Y_i, based on the received channel output, to the symbol node Y_i; and passing from the channel node R_z a likelihood of the symbol Z_k, based on the received channel output, to the symbol node Z_k (Figure 3 page 2 paragraph [0018]). Teaches of FIG. 5 and Xu teachings are analogous art because they are from the same field of endeavor. Even it is inherit in FIG. 5 at the time of the invention it would have been obvious to a person of ordinary skill in the art to combine the likelihood of the value as disclosed by Xu with the turbo decoder disclosed in FIG. 5. The suggestion/motivation for doing so would have been to determine when to stop the iteration process.

As per claim 16, admitted prior art in FIG. 5 shows a method for decoding a sequence of turbo code where the initializing outgoing messages from symbol nodes X_i, Y_i and Z_k includes: passing a message from the symbol node X_i to the computational node C_i of the computational node C, where the message is based on a summation of incoming messages at the symbol node X_i (FIG. 5 input 541 of block 501 with previous values input 542 of block 501); passing a message from the symbol node X_i to the computational node D_k of the computational node D, where the message is based on a summation of incoming messages at the symbol node X_i (FIG. 5 input 532 of block 502 with previous values input 540 of block 502); passing a message from the symbol node Y_i to the computational node C_i (FIG. 5 input 542 of block 501); and passing a message from the symbol node Z_k to the computational node D_k (FIG. 5 input 540 of block 502). It is inherit that passing a message from the symbol node Y_i to the computational node C_i is based in the likelihood of the data symbol (input 541 of block 501). It is inherit that passing a message from the symbol node Z_k to the computational node D_k is based in the likelihood of the data symbol (input 532 of block 502). This is very well known in turbo decoding process and Xu teaches that passing a message from the symbol node Y_i to the computational node C_i is based in the likelihood of the data symbol (Figure 3 L_a page 2 paragraph [0018]). It is inherit that passing a message from the symbol node Z_k to the computational node D_k is based in the likelihood of the data symbol (Figure 3 L_{e1} page 2 paragraph [0018]). Teaches of FIG. 5 and Xu teachings are analogous art because they are from the same field of endeavor. Even it is inherit in FIG. 5 at the time of the invention it would have been obvious to a person of ordinary skill in the art to combine the likelihood of the value as disclosed by Xu with the turbo decoder disclosed in FIG. 5. The suggestion/motivation for doing so would have been to determine when to stop the iteration process.

(10) Response to Argument

Applicant's arguments filed on 08/04/2006 have been fully considered but they are not persuasive.

Regarding Claim 1:

The Applicant contends, "We respectfully submit that the admitted prior art in Figure 5 does not describe, either expressly or inherently, the limitation of triggering all said computational nodes C and D at different instances of time essentially concurrently for each decoding iteration".

The Examiner disagrees and asserts, that, as indicated in the previous Office action, admitted prior art in FIG. 5 discloses that the triggering schedule includes triggering all the computational nodes C and D at different instances of time essentially concurrently for each decoding iteration (block 501 (computational node C), is the first decoder with inputs, essentially, X (information data) and Y (fist parity), when the first decoder finish, pass the output to block 502 (computational node D), that at a different instant in time, essentially concurrently, with inputs X (information data) and Z (second parity, or parity from de second encoder) produces the first estimation of the data in iteration zero).

This is specifically disclosed in the admitted prior art, indicating with respect at different instant of time "Decoder 502 produces estimates of data symbols X_k at an output 560. The decoding processes in decoders 501 and 502 may be performed sequentially" (page 11 lines 2-6), sequentially have to be performed at different instant of time.

With respect to essentially concurrently:

a) defined by the Applicant as all the computation are updated. Admitted prior art specifically discloses that "The information may pass from decoder 501 to decoder 502 after completing each iteration" (page 11 lines 4-5), so literally the information is updated in each iteration; it is also specifically disclosed by admitted prior art that "To increase confidence for the estimate of the data symbols X_i to hold a true value, the estimate of data symbols X_i at output 550 may pass through an interleaver 530 to produce estimates of data symbols X_k at an input 532 of decoder 502. Decoder 502 uses the estimates of data symbols X_k at input 532 with estimates of data symbols at input 540 to produce new estimates of data symbols X_k at output 560. Estimates of data symbols X_k at output 560 pass through a de-interleaver 531 to reverse the process of interleaving function of interleaver 430 in the turbo code 400, and to produce estimates of data symbols X_i at an input 541. Estimates of data symbols X_i at input 541 are used with the estimates of data symbols at input 542 to produce a new estimate of data symbols X_i at output 550. The process may be repeated until confidence for the estimate of data symbols X_i reaches an acceptable level", so the process is repeatedly updated in each iteration until an acceptable level, until it convergence to the desired point.

b) As defined by the American Heritage College Dictionary, used in the USPTO concurrently means "1. Happening at the same time as something else. 2. Operating or acting in conjunction with another. 3. Meeting or tending to meet the same point, convergent. 4. Being in accordance, harmonious". The definition of happening at the same time have to be disregarded, because it cannot happen at "different times" (see previous paragraph) and at the same time simultaneously; so all the other acceptations are clearly indicated in the turbo decoded where each decoder works in conjunction, in accordance and trying to meeting the same point (the decoded data) this was already presented by the Examiner in the last Office action, literally: “because the successive iterations “have to be convergent”, “they have to agree in opinion”, “they have to meet or tending to meet the same point”, “they can be processed at the same time that other processes” such as the D, “they can acting in conjunction with other processes” such as D, “meeting or tending to meet the same point”; “in accordance” and “in harmony” to the encoded data if not, instead of improving the results with each iteration the results will diverge and then decoder is not working because in each iteration produce a worst result that in the previous iteration, and the decoder will diverge”).

This is overwhelmingly well known by a person of ordinary skills in the art.

The Applicant contends, “there is nothing in either Figure 5 or the description of the decoder 500 that would specify the particular order in which the computational nodes are triggered within each of the constituent decoders 501 and 502 for each iteration”.

The Examiner disagrees and asserts, that, as indicated in the previous Office action, admitted prior art in FIG. 5 discloses that “To increase confidence for the estimate of the data symbols X_i to hold a true value, the estimate of data symbols X_i at output 550 may pass through an interleaver 530 to produce estimates of data symbols X_k at an input 532 of decoder 502. Decoder 502 uses the estimates of data symbols X_k at input 532 with estimates of data symbols at input 540 to produce new estimates of data symbols X_k at output 560. Estimates of data symbols X_k at output 560 pass through a de-interleaver 531 to reverse the process of interleaving function of interleaver 430 in the turbo code 400, and to produce estimates of data symbols X_i at an input 541. Estimates of data symbols X_i at input 541 are used with the estimates of data symbols at input 542 to produce a new estimate of data symbols X_i at output 550. The process may be repeated until confidence for the estimate of data symbols X_i reaches an acceptable level" (see also response to arguments above).

The Applicant contends, "Neither Figure 5 nor the description of the operation of the decoder 500 shown in' Figure 5 specifies the order of triggering of the computational nodes in each iteration, **with the exception that the decoding processes in decoders 501 and 502 may be performed sequentially**. In contrast, claim 1 requires triggering all said computational nodes C and D at different instances of time essentially concurrently for each decoding iteration" (emphasis added by the Examiner in bold).

The Examiner disagrees and asserts, that, as indicated in the previous Office action, admitted prior art in FIG. 5 discloses that "To increase confidence for the estimate of the data symbols X_i to hold a true value, the estimate of data symbols X_i at output 550 may pass through an interleaver 530 to produce estimates of data symbols X_k at an input 532 of decoder 502. Decoder 502 uses the estimates of data symbols X_k at input 532 with estimates of data symbols at input 540 to produce new estimates of data symbols X_k at output 560. Estimates of data symbols X_k at output 560 pass through a de-interleaver 531 to reverse the process of interleaving function of interleaver 430 in the turbo code 400, and to produce estimates of data symbols X_i at an input 541.

Estimates of data symbols X_i at input 541 are used with the estimates of data symbols at input 542 to produce a new estimate of data symbols X_i at output 550. The process may be repeated until confidence for the estimate of data symbols X_i reaches an acceptable level" (see also response to arguments above).

The Applicant contends, "Applicants respectfully submit that the admitted prior art in Figure 5 does not anticipate claim 1 at least because the admitted prior art does not disclose that the computational nodes are triggered essentially concurrently for each decoding iteration".

The Examiner disagrees and asserts, that, as indicated in the previous Office action, admitted prior art in FIG. 5 discloses "Referring to FIG. 5, a block diagram of a decoder 500 is shown for decoding the noisy version of data symbols Xi, Yi and Zk to produce estimates of data symbols Xi. Decoder 500 may be used in decoder 20 of system 10. The noisy version of encoded data symbols Xi, Yi and Zk may pass through a data symbol selector block 520 which operates to select the noisy version of data symbols Xi and Yi for routing to a decoder block 501 at an input 542. The noisy version of data symbols Xi internally passes through an interleaver 599 to locally reproduce a noisy version of data symbols Xk. The locally produced noisy version of data symbols Xk and Zk pass to a decoder block 502 at an input 540. Decoder 501 may decode the noisy version of data symbols Xi and Yi according to a decoding process such as MAP as explained and shown. Decoder 501 produces estimates of data symbols Xi at an output 550. Decoder 502 decodes the noisy version of data symbols Zk and Xk according to a decoding process such as MAP as explained and shown. Decoder 502 produces estimates of data symbols Xk at an output 560. The decoding processes in decoders 501 and 502 may be performed sequentially. The information may pass from decoder 501 to decoder 502 after completing each iteration. One ordinary skilled in the art may appreciate that the decoders in various embodiments as described and shown operate on the noisy version of the encoded data symbols. To increase confidence for the estimate of the data symbols X_i to hold a true value, the estimate of data symbols X_i at output 550 may pass through an interleaver 530 to produce estimates of data symbols X_k at an input 532 of decoder 502. Decoder 502 uses the estimates of data symbols X_k at input 532 with estimates of data symbols at input 540 to produce new estimates of data symbols X_k at output 560. Estimates of data symbols X_k at output 560 pass through a de-interleaver 531 to reverse the process of interleaving function of interleaver 430 in the turbo code 400, and to produce estimates of data symbols X_i at an input 541. Estimates of data symbols X_i at input 541 are used with the estimates of data symbols at input 542 to produce a new estimate of data symbols X_i at output 550. The process may be repeated until confidence for the estimate of data symbols X_i reaches an acceptable level" (emphasis added by the Examiner) (see also response to arguments above).

For these reasons and the reasons indicated in the previous Office action the rejection of claim 1 is maintained.

Regarding Claim 6:

The Applicant contends, "As discussed in more detail above in relation to claim 1, Figure 5 and its description do not teach concurrent triggering of multiple computational nodes. Furthermore, Figure 5 and its description fail to disclose the step of partitioning. At least for these reasons, Applicants respectfully submit that the admitted prior art of Figure 5 does not anticipate independent claim 6".

The Examiner disagrees and asserts, that, as indicated in the previous Office action, and as indicated in the respond to argument regarding claim 1, admitted prior art in FIG. 5 discloses that "To increase confidence for the estimate of the data symbols X_i to hold a true value, the estimate of data symbols X_i at output 550 may pass through an interleaver 530 to produce estimates of data symbols X_k at an input 532 of decoder 502. Decoder 502 uses the estimates of data symbols X_k at input 532 with estimates of data symbols at input 540 to produce new estimates of data symbols X_k at output 560. Estimates of data symbols X_k at output 560 pass through a de-interleaver 531 to reverse the process of interleaving function of interleaver 430 in the turbo code 400, and to produce estimates of data symbols X_i at an input 541. Estimates of data symbols X_i at input 541 are used with the estimates of data symbols at input 542 to produce a new estimate of data symbols X_i at output 550. The process may be repeated until confidence for the estimate of data symbols X_i reaches an acceptable level" (see also response to arguments above).

Al so as indicated in the previous Office action, In Figure 5 for example the first subset could be C_0 and C_1 (first decoder in iteration zero and first decoder in iteration 1), the second subset could be C_1 and C_2 (first decoder in iteration 1 and first decoder in iteration 2), etc.. so divided into several overlapping sub-blocks, and the nodes are triggered sequentially within each sub-block (iteration 1 needs to wait for iteration zero, because the input of iteration 1 need that iteration zero to be performed previously), but concurrently across all sub-blocks, because the successive iterations "have to be convergent", "they have to agree in opinion", "they have to meet or tending to meet the same point", "they can be processed at the same time that other processes" such as the D, "they can acting in conjunction with other processes" such as D, "meeting or tending to meet the same point"; "in accordance" and "in harmony" to the encoded data if not, instead of improving the results with each iteration the results will diverge and then decoder is not working because in each iteration produce a worst result that in the previous iteration, and the decoder will diverge.

For these reasons and the reasons indicated in the previous Office action the rejection of claim 6 is maintained.

Regarding Claims 18 and 22:

The Applicant contends, "these claims therefore also requires concurrent triggering of a plurality of computational nodes. As discussed in more detail above in relation to claim 1, the admitted prior art of Figure 5 does not disclose concurrent triggering of a plurality of computational nodes. At least for this reason, Applicants respectfully submit that the admitted prior art of Figure 5 does not anticipate independent claims 18 and 22".

The Examiner disagrees and asserts, that, as indicated in the previous Office action, and as indicated with more detail in the respond to arguments of claims 1 and 6 previously, admitted prior art in FIG. 5 shows teaches "Referring to FIG. 5, a block diagram of a decoder 500 is shown for decoding the noisy version of data symbols Xi, Yi and Zk to produce estimates of data symbols Xi. Decoder 500 may be used in decoder 20 of system 10. The noisy version of encoded data symbols Xi, Yi and Zk may pass through a data symbol selector block 520 which operates to select the noisy version of data symbols Xi and Yi for routing to a decoder block 501 at an input 542. The noisy version of data symbols Xi internally passes through an interleaver 599 to locally reproduce a noisy version of data symbols Xk. The locally produced noisy version of data symbols Xk and Zk pass to a decoder block 502 at an input 540. Decoder 501 may decode the noisy version of data symbols Xi and Yi according to a decoding process such as MAP as explained and shown. Decoder 501 produces estimates of data symbols Xi at an output 550. Decoder 502 decodes the noisy version of data symbols Zk and Xk according to a decoding process such as MAP as explained and shown. Decoder 502 produces estimates of data symbols Xk at an output 560. The decoding processes in decoders 501 and 502 may be performed sequentially. The information may pass from decoder 501 to decoder 502 after completing each iteration. One ordinary skilled in the art may appreciate that the decoders in various embodiments as described and shown operate on the noisy version of the encoded data symbols. To increase confidence for the estimate of the data symbols X_i to hold a true value, the estimate of data symbols X_i at output 550 may pass through an interleaver 530 to produce estimates of data symbols X_k at an input 532 of decoder 502. Decoder 502 uses the estimates of data symbols X_k at input 532 with estimates of data symbols at input 540 to produce new estimates of data symbols X_k at output 560. Estimates of data symbols X_k at output 560 pass through a de-interleaver 531 to reverse the process of interleaving function of interleaver 430 in the turbo code 400, and to produce estimates of data symbols X_i at an input 541. Estimates of data symbols X_i at input 541 are used with the estimates of data symbols at input 542 to produce a new estimate of data symbols X_i at output 550. The process may be repeated until confidence for the estimate of data symbols X_i reaches an acceptable level" (emphasis added by the Examiner) (see also response to arguments above), so, as indicated in the previous Office action, the two decoders being triggered concurrently, because the successive iterations "have to be convergent", "they have to agree in opinion", "they have to meet or tending to meet the same point", if not, instead of improving the results with each iteration the results will diverge and then decoder is not working because in each iteration produce a worst result that in the previous iteration, and the decoder will diverge.

For these reasons and the reasons indicated in the previous Office action the rejection of claims 18 and 22 are maintained.

Dependent Claims:

The Applicant contends, "The above discussion addresses rejections of all independent claims of the application. Dependent claims should be patentable at least for the same reasons as their base claims and intervening claims, if any".

The Examiner disagrees and asserts, as indicated in the previous office Action, because the rejections of the independent claims is maintained, the rejections of the dependent claims is also maintained.

For these reasons and the reasons indicated in the previous Office action the rejection of other claims are maintained.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Juan Alberto Torres

08-23-2006

Conferees:

Mohammed Ghayour

Chieh Fan

MOHAMMED GHAYOUR
SUPERVISORY PATENT EXAMINER

CHIEH M. FAN
SUPERVISORY PATENT EXAMINER